

# Open-Source Software Basics

Stefanos Baziotis, [sdi1600105@di.uoa.gr](mailto:sdi1600105@di.uoa.gr), [@baziotis](#) (Github)

# What is Open-Source Software?

- Simply, software for which the source code is available
- However...

# What is Open-Source Software?

- Simply, software for which the source code is available
- However... The fact that some software is open-source, usually implies a couple of things about its development.



# How OSS Development Usually Works

- The software is developed by a community.

# How OSS Development Usually Works

- The software is developed by a community.
- This starts with someone that publishes some code that programs some software (and usually, only the source code).



# How OSS Development Usually Works

- The software is developed by a community.
- This starts with someone that publishes some code that programs some software (and usually, only the source code).
- Then people interested in this software come and *contribute* to its development.

# How OSS Development Usually Works

- The software is developed by a community.
- This starts with someone that publishes some code that programs some software (and usually, only the source code).
- Then people interested in this software come and *contribute* to its development.
- This community-driven development is what OSS is usually about.



# How OSS Development Usually Works

- **Step 1:** Someone wants to publish their work.
- *Do they just upload a folder / directory with their work somewhere online ?*



# How OSS Development Usually Works

- **Step 1:** Someone wants to publish their work.
- *Do they just upload a folder / directory with their work somewhere online ?*
- Usually No and for a good reason.
- We need some more complex capabilities that simple folders don't provide.

# Source Code Publishment

- This is where **Git** comes into play.



# Source Code Publishment

- This is where **Git** comes into play.
- The basic capability that it provides is the concept of history.

# Source Code Publishment

- This is where **Git** comes into play.
- The basic capability that it provides is the concept of history.
- **Go to Live Demo Mode!**



# Source Code Publishment

- Git works with repositories.
- A **Git repository** is just a fancy name for:
- „A folder with the current snapshot of my code along with all the files that **Git** needs“.
- So, what I publish, is a **Git repository**.

# Source Code Publiishment

- *Ok, but where do I puslish this ? Do I upload that somewhere and then just send you the link ?*



# Source Code Publishment

- *Ok, but where do I publish this ? Do I upload that somewhere and then just send you the link ?*
- Again, usually No.

# Source Code Publishment

- This is where **Github** comes into play.
- It is an online platform for **Git** repositories.



# Source Code Publishment

- This is where **Github** comes into play.
- It is an online platform for **Git repositories**. Essentially, it provides a nice way to view a git repo.
- **Go to Live Demo Mode!**

# How Do I Contribute ?

- Step 2: The Contribution
- *Alright, that's all good, but how do I contribute to a repository ?*



# How Do I Contribute ?

- *Alright, that's all good, but how do I contribute to a repository ?*
- You **fork** the repository, you **commit** your changes and then **PR** those changes. A **PR review** follows and if the changes are welcome, they get **merged**.

# How Do I Contribute ?

- Let's explain each unknown term in the previous slide separately.



# The Fork

- A **fork** essentially means „A copy of a repository“.

# The Fork

- A **fork** essentially means „A copy of a repository“.
- That copied repository is owned by you (but it's visible that it is a copy of the original).



# The Fork

- You can now work on your copy independently.

# The Fork

- You can now work on your copy independently.
- Then when you're ready, you **PR** those changes.



# The Pull Request (PR)

- A **pull request (or PR)** is a *request* to the maintainer of the original repository to *pull* your changes.
- Pull here means just „incorporate my changes to your project“.

# The PR Review

- A PR review is, as the name suggests, a review of your PR, i.e. the changes that you proposed.



# The PR Review

- A PR review is, as the name suggests, a review of your PR, i.e. the changes that you proposed.
- This definitely includes a **code review** but also a review of why this change is beneficial, what are the intentions etc.

# The Merge

- Finally, if people approve the request, it will get **merged**. In the Git parlance, merging code involves a lot of details in which we won't get into.



# The Merge

- Finally, if people approve the request, it will get **merged**. In the Git parlance, merging code involves a lot of details in which we won't get into.
- Bottom-line, your changes were accepted and are now part of the original repo.

# Other Kinds of Contributions

- In essence, the reason that one contributes to OSS is because they want to see better software in the world.



# Other Kinds of Contributions

- In essence, the reason that one contributes to OSS is because they want to see better software in the world.
- For good software to exist, we not only need people that write good code!

# Other Kinds of Contributions

- We need people that know how to **document** (i.e explain) code.



# Other Kinds of Contributions

- We need people that know how to **document** (i.e explain) code.
- We need people that **find bugs** (i.e. problems) in the software.

# Other Kinds of Contributions

- We need people that know how to **document** (i.e explain) code.
- We need people that **find bugs** (i.e. problems) in the software.
- People that are good at **helping** others get better at programming.



# Other Kinds of Contributions

- And let's not forget the **soft skills**. Remember that communities play a central role in OSS. And directing a healthy community is harder than it may sound.

# The Key Take-away

- We need people that understand what good software feels like, they want to see more such software and they are willing to help in its development!



# Benefits

- *What OSS offers to me ?*
- Meaningful OSS contributions are one of the most important kinds of experience to include in a CV.

# Benefits

- *What OSS offers to me ?*
- You get to **become a better programmer**! Although contributing to OSS is not necessary to become a good programmer, it is potentially sufficient.



# Benefits

- *What OSS offers to me ?*
- You get to interact with some of the best programmers in the world. This is related to the previous one but it obviously stands on its own. :)

# Benefits

- *What OSS offers to me ?*
- You get to **program** not because someone told you to but **because you chose to..!**



# How we will operate

- We will communicate via [Slack](#).
- This is a forum-like service that can help us have public discussions.

# How we will operate

- We will communicate via [Slack](#).
- This is a forum-like service that can help us have public discussions.
- Other than that, 4 main actions are proposed to help people get involved in open-source development.



# The 4 Actions

1. A mock project
2. Talks from you in technical topics (e.g. Git usage)
3. Talks from you in either your own open-source project or a presentation of a popular one.

# The 4 Actions

4. Questions and Answers in Slack.



# How to join us

Use this [invitation link](#) to join us in Slack.

# How to join us

Use this [invitation link](#) to join us in Slack.  
Then, join the #open\_source channel.



# How to join us

Use this [invitation link](#) to join us in Slack.  
Then, join the #open\_source channel.

# Thank you!